Implement three nodes point–to–point network with duplex links between them. Set the queue size, vary the bandwidth and find the number of packets dropped.

```
#-------Event scheduler object creation--------#
set ns [new Simulator]

#----------creating nam objects----------------#
set nf [open tcp1.nam w]
$ns namtrace-all $nf

#open the trace file
set nt [open tcp1.tr w]
$ns trace-all $nt

#------creating client- router- end server node------#
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

#-------- node colors---------------------#
$n0 color blue
$n1 color red
$n2 color green

#-------- Data flow color-------------#
$ns color 0 purple

#---creating duplex link---------#
$ns duplex-link $n0 $n1 1Mb 100ms DropTail
$ns duplex-link $n1 $n2 600Kb 100ms DropTail


#setting queue size of the link
$ns queue-limit $n1 $n2 5


#------creating orientation-------#
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right

#------------Labelling----------------#
$ns at 0.0 "$n0 label Client1"
$ns at 0.0 "$n1 label Router1"
$ns at 0.0 "$n2 label Endserver1"

#-----------Configuring nodes------------#
$n2 shape hexagon
$n1 shape square
```

```
#creating a udp connection in network simulator
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

#set up CBR over udp
set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize_ 800
$cbr0 set interval_ 0.008
$cbr0 attach-agent $udp0

set sink [new Agent/Null]
$ns attach-agent $n2 $sink
$ns connect $udp0 $sink


#---------finish procedure--------#
proc finish {} {
         global ns nf nt
         $ns flush-trace
         close $nf
         close $nt

         puts "running nam..."
         exec nam tcp1.nam &
         exit 0
      }

#scheduling events
$ns at 0.1 "$cbr0 start"
$ns at 3.0 "$cbr0 stop"
$ns at 3.5 "finish"

$ns run
```